Solving an SVM by hand based on KKT conditions

by Forrest Sheng Bao at Iowa State University
Copyleft 2020

Although necessary but not sufficient, the KKT conditions are sufficient for optimality if the problem is to minimize a convex function which is the case for an SVM in primal form. Therefore, from KKT conditions alone, we can solve an SVM by hand.

Warm-up: Solving a system of equations in Mathematica

In[ ]:=
```
Solve[{a1 + a2 == 2, a1 - a2 == 10}, {a1, a2}]
```

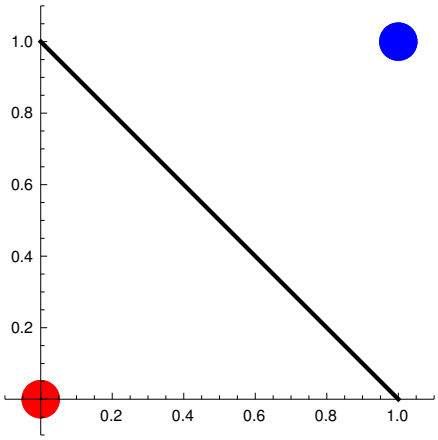
Out[ ]=
```
{{a1 → 6, a2 → -4}}
```

Example 1: Just two samples, one for each class.

The plot below indicates two two samples, (1,1) of class +1, and (0,0) of class -1. Denote the first one as sample 1 and the second as sample 2. A well-trained SVM would be the one depicted by the blue line.

In[96]:=
```
plt1 = ListPlot[{{0, 0}}, PlotStyle → {AbsolutePointSize[20], Red},
   AspectRatio → 1, PlotRange → {{-0.1, 1.1}, {-0.1, 1.1}}];
plt2 = ListPlot[{{1, 1}}, PlotStyle → {AbsolutePointSize[20], Blue},
   AspectRatio → 1, PlotRange → All];
plt3 = ListLinePlot[{{0, 1}, {1, 0}}, PlotRange → All, PlotStyle → {Thickness[.01], Black}];
Show[plt1, plt2, plt3]
```

Out[99]=



1. First, using the gradient on the bias term that $\sum_{k=1}^{K} \lambda_k \, yk = 0$, we have $(\lambda_1 \quad \lambda_2)\begin{pmatrix} +1 \\ -1 \end{pmatrix} = 0$
2. Then, write one equation for each sample in the complementary slackness equation (Equation D in slides):
   $\lambda_1\left[+1\cdot\left((w1 \quad w2)\begin{pmatrix} 1 \\ 1 \end{pmatrix} + w_b\right) - 1\right] = 0$
   $\lambda_2\left[-1\cdot\left((w1 \quad w2)\begin{pmatrix} 0 \\ 0 \end{pmatrix} + w_b\right) - 1\right] = 0$
3. Finally, the gradient on the non-bias weights: $\begin{pmatrix} w1 \\ w2 \end{pmatrix} = \lambda_1 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \lambda_2 \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Expand matrix operations above into a system of equations and solve them using the Solve function. For typesetting

easiness, λ's are replaced with a's in the code below.

In[ ]:=
```
Solve[{ a1 - a2 == 0,
        a1 * (w1 + w2 + wb - 1) == 0,
        a2 * (-1 * wb - 1) == 0,
        w1 == a1,
       w2 == a1
 }, {a1, a2, w1, w2, wb}
]
```

Solve: Equations may not give solutions for all "solve" variables.

Out[ ]= {{a1 → 0, a2 → 0}, {a1 → 0, a2 → 0, wb → -1}, {a1 → 0, a2 → 0, wb → 1}, {a1 → 1, a2 → 1, wb → -1}}

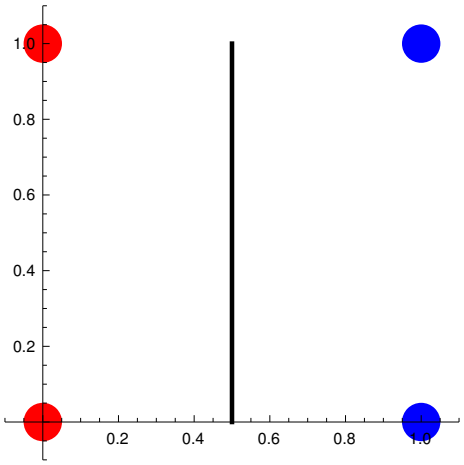
The first 3 solutions set both a1 and a2 to 0. They should be discarded as they remove the constraints from the Lagrange multiplier.
The last solution is what we want, demonstrating an equation $x + y - 1 = 0$ which beautifully goes between the two samples at 135 degree. The fact that neither a1 nor a2 are zero indicates that both samples are the supporting vectors.

Example 2

Class +1: (0,0) and (0,1)
Class -1: (1,0) and (1,1).

In[88]:=
```
plt1 = ListPlot[{{0, 0}, {0, 1}}, PlotStyle → {AbsolutePointSize[20], Red},
   AspectRatio → 1, PlotRange → {{-0.1, 1.1}, {-0.1, 1.1}}];
plt2 = ListPlot[{{1, 0}, {1, 1}}, PlotStyle → {AbsolutePointSize[20], Blue},
   AspectRatio → 1, PlotRange → All];
plt3 = ListLinePlot[{{0.5, 0}, {0.5, 1}}, PlotRange → All, PlotStyle → {Thickness[.01], Black}];
Show[plt1, plt2, plt3]
```

Out[91]=



In[3]:=
```
Solve[{a1 + a2 - a3 - a4 == 0,
       a1 * (1 * (w1 + w2 + wb) - 1) == 0,
       a2 * (1 * (w1 + 0 + wb) - 1) == 0,
       a3 * (-1 * (0 + 0 + wb) - 1) == 0,
       a4 * (-1 * (0 + w2 + wb) - 1) == 0,
       w1 == a1 + a2,
       w2 == a1 - a4,
  a1 ≥ 0, a2 ≥ 0, a3 ≥ 0, a4 ≥ 0
 }, {w1, w2, a1, a2, a3, a4, wb}
]
```

Solve: Equations may not give solutions for all "solve" variables.

Out[3]=
```
{{w1 → 0, w2 → 0, a1 → 0, a2 → 0, a3 → 0, a4 → 0},
 {w1 → ConditionalExpression[2, 0 < a4 < 2], w2 → ConditionalExpression[0, 0 < a4 < 2],
  a1 → ConditionalExpression[a4, 0 < a4 < 2], a2 → ConditionalExpression[2 - a4, 0 < a4 < 2],
  a3 → ConditionalExpression[2 - a4, 0 < a4 < 2], wb → ConditionalExpression[-1, 0 < a4 < 2]},
 {w1 → 1, w2 → -1, a1 → 0, a2 → 1, a3 → 0, a4 → 1, wb → 0},
 {w1 → 1, w2 → 1, a1 → 1, a2 → 0, a3 → 1, a4 → 0, wb → -1},
 {w1 → 2, w2 → 0, a1 → 0, a2 → 2, a3 → 2, a4 → 0, wb → -1},
 {w1 → 2, w2 → 0, a1 → 2, a2 → 0, a3 → 0, a4 → 2, wb → -1}}
```

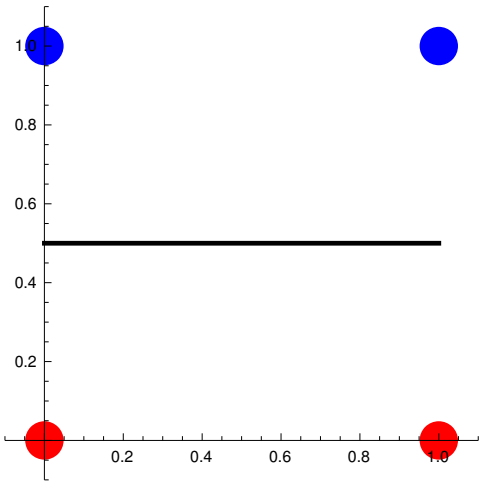
Example 3

Class + 1 : (1, 1) and (0, 1)

Class - 1 : (0, 0) and (1, 0).

In[80]:=
```
plt1 = ListPlot[{{0, 0}, {1, 0}}, PlotStyle → {AbsolutePointSize[20], Red},
   AspectRatio → 1, PlotRange → {{-0.1, 1.1}, {-0.1, 1.1}}];
plt2 = ListPlot[{{0, 1}, {1, 1}}, PlotStyle → {AbsolutePointSize[20], Blue},
   AspectRatio → 1, PlotRange → All];
plt3 = ListLinePlot[{{0, 0.5}, {1, 0.5}}, PlotRange → All, PlotStyle → {Thickness[.01], Black}];
Show[plt1, plt2, plt3]
```

Out[83]=



In[4]:=
```
Solve[{a1 - a2 - a3 + a4 == 0,
       a1 * (1 * (w1 + w2 + wb) - 1) == 0,
       a2 * (-1 * (w1 + 0 + wb) - 1) == 0,
       a3 * (-1 * (0 + 0 + wb) - 1) == 0,
       a4 * (1 * (0 + w2 + wb) - 1) == 0,
       w1 == a1 - a2,
       w2 == a1 + a4,
  a1 ≥ 0, a2 ≥ 0, a3 ≥ 0, a4 ≥ 0
 }, {w1, w2, a1, a2, a3, a4, wb}
]
```

Solve: Equations may not give solutions for all "solve" variables.

Out[4]=

$$\left\{\{w1 \to 0, w2 \to 0, a1 \to 0, a2 \to 0, a3 \to 0, a4 \to 0\},\right.$$

$$\left\{w1 \to \text{ConditionalExpression}\left[\frac{-2\,(2-a4)+2\left(2-\frac{4-2\,(2-a4)-2\,a4}{2-a4}-a4\right)-2\,a4+2\left(\frac{4-2\,(2-a4)-2\,a4}{2-a4}+a4\right)}{2-a4}, 0<a4<2\right],\right.$$

$$w2 \to \text{ConditionalExpression}[2, 0<a4<2], a1 \to \text{ConditionalExpression}[2-a4, 0<a4<2],$$

$$a2 \to \text{ConditionalExpression}\left[2-\frac{4-2\,(2-a4)-2\,a4}{2-a4}-a4, 0<a4<2\right],$$

$$a3 \to \text{ConditionalExpression}\left[\frac{4-2\,(2-a4)-2\,a4}{2-a4}+a4, 0<a4<2\right],$$

$$\left.wb \to \text{ConditionalExpression}[-1, 0<a4<2]\right\}, \{w1 \to -1, w2 \to 1, a1 \to 0, a2 \to 1, a3 \to 0, a4 \to 1, wb \to 0\},$$

$$\{w1 \to 0, w2 \to 2, a1 \to 0, a2 \to 0, a3 \to 2, a4 \to 2, wb \to -1\},$$

$$\{w1 \to 0, w2 \to 2, a1 \to 2, a2 \to 2, a3 \to 0, a4 \to 0, wb \to -1\},$$

$$\left.\{w1 \to 1, w2 \to 1, a1 \to 1, a2 \to 0, a3 \to 1, a4 \to 0, wb \to -1\}\right\}$$